



Adicionando Push Notification

Overview sobre Firebase

O Firebase é considerado uma plataforma de aplicação web. Isso ajuda os desenvolvedores a criarem aplicativos de alta qualidade. Ele armazena os dados em formato JavaScript Object Notation (JSON) que não usa consulta para inserir, atualizar, excluir ou adicionar dados a ele. Isso é o backend de um sistema que é usado como banco de dados para armazenar dados.

Os serviços disponíveis são:

Firebase Analytics

Ele fornece informações sobre o uso do aplicativo. É um aplicativo pago de medição que também proporciona engajamento do usuário. Esse recurso exclusivo permite que o desenvolvedor de aplicativos entenda como os usuários estão usando o aplicativo. O SDK tem o recurso de capturar eventos e propriedades por conta própria e também permite obter dados personalizados.

Firebase Cloud Messaging (FCM)

Anteriormente conhecido como Google Clouds Messaging (GCM), o FCM é um serviço pago que é uma solução multiplataforma para mensagens e notificações para Android, aplicativos da Web, e IOS.

Autenticação do Firebase

O Firebase Auth oferece suporte a provedores de login social como o Facebook, Google GitHub e Twitter. É um serviço que pode autenticar usuários usando apenas código do lado do cliente e é pago pelo serviço. Inclui também um sistema de gestão de utilizadores através do qual os desenvolvedores podem habilitar a autenticação do usuário com e-mail e login de senha armazenados no Firebase.

Banco de dados em tempo real.

O Firebase fornece serviços como banco de dados em tempo real e processo interno. Uma API é fornecida ao desenvolvedor do aplicativo que permite que os dados do aplicativo sejam sincronizados entre clientes e armazenados na nuvem do Firebase. As bibliotecas cliente são fornecidas pela empresa que permite a integração com Aplicativos Android, IOS e JavaScript.

Armazenamento do Firebase

Facilita a transferência de arquivos fácil e segura, independentemente da rede qualidade para os aplicativos do Firebase. É apoiado pelo Google Cloud armazenamento que é um serviço de armazenamento de objetos econômico. O desenvolvedor pode usá-lo para armazenar imagens, áudio, vídeo ou outros conteúdos gerados por usuários.

Laboratório de teste do Firebase para Android

Ele fornece infraestrutura baseada em nuvem para testar o Android aplicativos. Com uma operação, os desenvolvedores podem iniciar o teste de seus aplicativos em uma ampla variedade de dispositivos e configurações. Os vários resultados de teste, como capturas de tela, vídeos e registros estão disponíveis no console do Firebase. Mesmo

que um desenvolvedor não tenha escrito nenhum código de teste para seu aplicativo, Test Lab pode exercitar o aplicativo automaticamente, procurando por falhas.

Relatório de falhas do Firebase

Os relatórios detalhados dos erros são criados no aplicativo. Os erros são agrupados em clusters de rastreamentos de pilha semelhantes e triados pela gravidade. Os outros recursos são: o desenvolvedor pode registrar eventos personalizados para ajudar a capturar as etapas que levam a uma batida.

Notificações do Firebase

Ele permite notificações de usuários direcionadas para aplicativos móveis desenvolvidos e os serviços estão disponíveis gratuitamente.

Criação do Projeto para Push Notification

Um dos recursos mais comuns fornecidos pelos desenvolvedores de aplicativos a seus usuários são as notificações push.

Para fins de registro e monitoramento de notificações push do Firebase, usaremos o API de notificação por push para capacitar uma aplicação Ionic + Angular.

Dependências Necessárias

Construir e implantar aplicativos iOS e Android usando o Capacitor requer um pouco de configuração.

Para testar notificações push no iOS, a Apple exige que você tenha uma conta de desenvolvedor Apple paga e um dispositivo iOS físico .

Se você estiver com problemas ou seu console emitir avisos sobre pacotes desatualizados ou obsoletos, verifique se você e nas versões estáveis mais recentes do Node, Android Studio e Xcode.

Além disso, estamos usando o Firebase para notificações push, portanto, se você estiver usando outros plug-ins do Cordova que usam o SDK do Firebase, verifique se eles estão usando as versões mais recentes.

Prepare um aplicativo de capacitor iônico

Vamos criar um aplicativo Ionic primeiro.

Em seu terminal preferido, instale a versão mais recente do Ionic CLI:

npm install -g @ionic/cli

Em seguida, vamos usar a CLI para criar um novo aplicativo Ionic Angular com base no projeto inicial em branco e chamá-lo de capApp :

ionic start capApp blank --type=angular

No prompt solicitando a integração do seu novo aplicativo com o Capacitor, digite ye pressione enter. Isso adiciona o Capacitor e o Capacitor CLI ao nosso novo aplicativo.

Depois que o aplicativo for criado com sucesso, alterne para o diretório do projeto recém-criado:

cd capApp/

Finalize executando npx cap init, o que nos permitirá preencher as informações do nosso aplicativo.

npx cap init



? App name: CapApp

? App Package ID: com.mydomain.myappname

Criando o aplicativo e adicionando plataformas

Antes de adicionar qualquer plataforma nativa a este projeto, o aplicativo deve ser compilado pelo menos uma vez. Uma compilação da web cria o diretório de ativos da web que o Capacitor precisa (www pasta em projetos Ionic Angular).

ionic build

Em seguida, vamos adicionar as plataformas iOS e Android ao nosso aplicativo.

npx cap add ios

npx cap add android

Ao executar esses comandos, ambas as pastas android e ios na raiz do projeto são criadas. Esses são artefatos de projeto nativos totalmente separados que devem ser considerados parte do seu aplicativo Ionic (ou seja, verifique-os no controle de origem).

Usando a API de notificação por push do capacitor

Primeiro de tudo, precisamos instalar o Plugin de Notificações Push do Capacitor:

npm install @capacitor/push-notifications

npx cap sync

Então, antes de chegarmos ao Firebase, precisaremos gara que nosso aplicativo possa se registrar para notificações push usando a API Capacitor Push Notification. Também adicionaremos um alert(você pode usar console.log instruções) para nos mostrar a carga útil de uma notificação quando ela chegar e o aplicativo estiver aberto em nosso dispositivo.

Em seu aplicativo, vá para o home.page.ts arquivo e adicione uma import instrução e um const para usar a API do Capacitor Push:

```
import {

  ActionPerformed,

  PushNotificationSchema,

  PushNotifications,

  Token,

} from '@capacitor/push-notifications';
```

Em seguida, adicione o ngOnInit() método com alguns métodos de API para registrar e monitorar notificações por push. Também adicionaremos alert() alguns dos eventos para monitorar o que está acontecendo:

```
export class HomePage implements OnInit {

ngOnInit() {

   console.log('Initializing HomePage');

   // Request permission to use push notifications
```

```
// iOS will prompt user and return if they granted permissior ⓘ ıot

// Android will just grant without prompting

PushNotifications.requestPermissions().then(result => {

 if (result.receive = 'granted') {

  // Register with Apple / Google to receive push via APNS/FCM

  PushNotifications.register();

 } else {

  // Show some error

 }

});

// On success, we should be able to receive notifications

PushNotifications.addListener('registration',

 (token: Token) => {

  alert('Push registration success, token: ' + token.value);

 }

);

// Some issue with our setup and push will not work

PushNotifications.addListener('registrationError',
```



```
  (error: any) => {
    alert('Error on registration: ' + JSON.stringify(error));
  }
);

// Show us the notification payload if the app is open on our device
PushNotifications.addListener('pushNotificationReceived',
  (notification: PushNotificationSchema) => {
    alert('Push received: ' + JSON.stringify(notification));
  }
);

// Method called when tapping on a notification
PushNotifications.addListener('pushNotificationActionPerformed',
  (notification: ActionPerformed) => {
    alert('Push action performed: ' + JSON.stringify(notification));
  }
);
}
```

Aqui está a implementação completa de home.page.ts:



```
import { Component, OnInit } from '@angular/core';

import {

 ActionPerformed,

 PushNotificationSchema,

 PushNotifications,

 Token,

} from '@capacitor/push-notifications';

@Component({

 selector: 'app-home',

 templateUrl: 'home.page.html',

 styleUrls: ['home.page.scss'],

})

export class HomePage implements OnInit {

 ngOnInit() {

  console.log('Initializing HomePage');

  // Request permission to use push notifications

  // iOS will prompt user and return if they granted permission or not

  // Android will just grant without prompting
```

```
PushNotifications.requestPermissions().then(result => {

 if (result.receive = 'granted') {

  // Register with Apple / Google to receive push via APNS/FCM

  PushNotifications.register();

 } else {

  // Show some error

 }

});

PushNotifications.addListener('registration', (token: Token) => {

 alert('Push registration success, token: ' + token.value);

});

PushNotifications.addListener('registrationError', (error: any) => {

 alert('Error on registration: ' + JSON.stringify(error));

});

PushNotifications.addListener(

 'pushNotificationReceived',

 (notification: PushNotificationSchema) => {

  alert('Push received: ' + JSON.stringify(notification));
```



```
    },
  );
  PushNotifications.addListener(
    'pushNotificationActionPerformed',
    (notification: ActionPerformed) => {
      alert('Push action performed: ' + JSON.stringify(notification));
    },
  );
 }
}
```

Depois disso, você deseja gerar uma nova compilação e informar o Capacitor sobre as alterações. Você pode fazer isso com:

ionic build

npx cap copy

Agora, a parte divertida - vamos verificar se as notificações push do Firebase estão funcionando no Android e no iOS!

Precisamos iniciar nosso aplicativo no Android ou iOS para que nossa home.page.ts página possa se cadastrar e receber notificações.

Para abrir seu projeto Android no Android Studio:

npx cap open android



Para abrir seu projeto iOS no Xcode:

npx cap open ios

Quando o projeto estiver aberto, carregue o aplicativo no seu dispositivo usando o recurso Executar do Android Studio ou do Xcode. O aplicativo deve iniciar na página inicial.

Observação: no iOS, você verá um pop-up solicitando que você permita notificações para seu aplicativo - certifique-se de escolher Permitir notificações!

Se o seu aplicativo for registrado com sucesso e você seguiu o código acima, você deverá ver um alerta com uma mensagem de sucesso!

Agora vamos testar para ver se as notificações são recebidas pelo nosso dispositivo. Para enviar uma notificação, no Firebase, acesse a seção Cloud Messaging no cabeçalho Grow no painel do projeto.

Em seguida, selecione o botão Nova notificação .

Ao criar a notificação, você só precisa especificar as seguintes informações:

1. O texto da notificação

2. O título (somente Android, opcional para iOS)

3. O Alvo (seja um segmento de usuário ou tópico; recomendo apenas segmentar o próprio aplicativo iOS ou Android)

4. O agendamento (deixe isso para "Agora")



Nesse ponto, você pode revisar a notificação que reuniu e selecionar Publicar para enviar a notificação.

Se você configurou seu aplicativo corretamente, verá um alerta pop-up na tela inicial com a notificação por push composta no Firebase. Você pode então tocar na notificação e deve receber um alert para o pushActionPerformed evento, de acordo com nosso código acima.

Implementando Push Notification com Angular

Para podermos configurar notificações push em Angular temos que começar abrindo seu terminal na pasta do projeto e executar o seguinte comando para instalar o pacote OneSignal NPM (OneSignal-ngx) .

```
npm i onesignal-ngx
```

Figura1: Instalação do pacote OneSignal

Agora, localize os arquivos SDK que você baixou no seu computador e insira-os dentro da src pasta do seu aplicativo Angular.

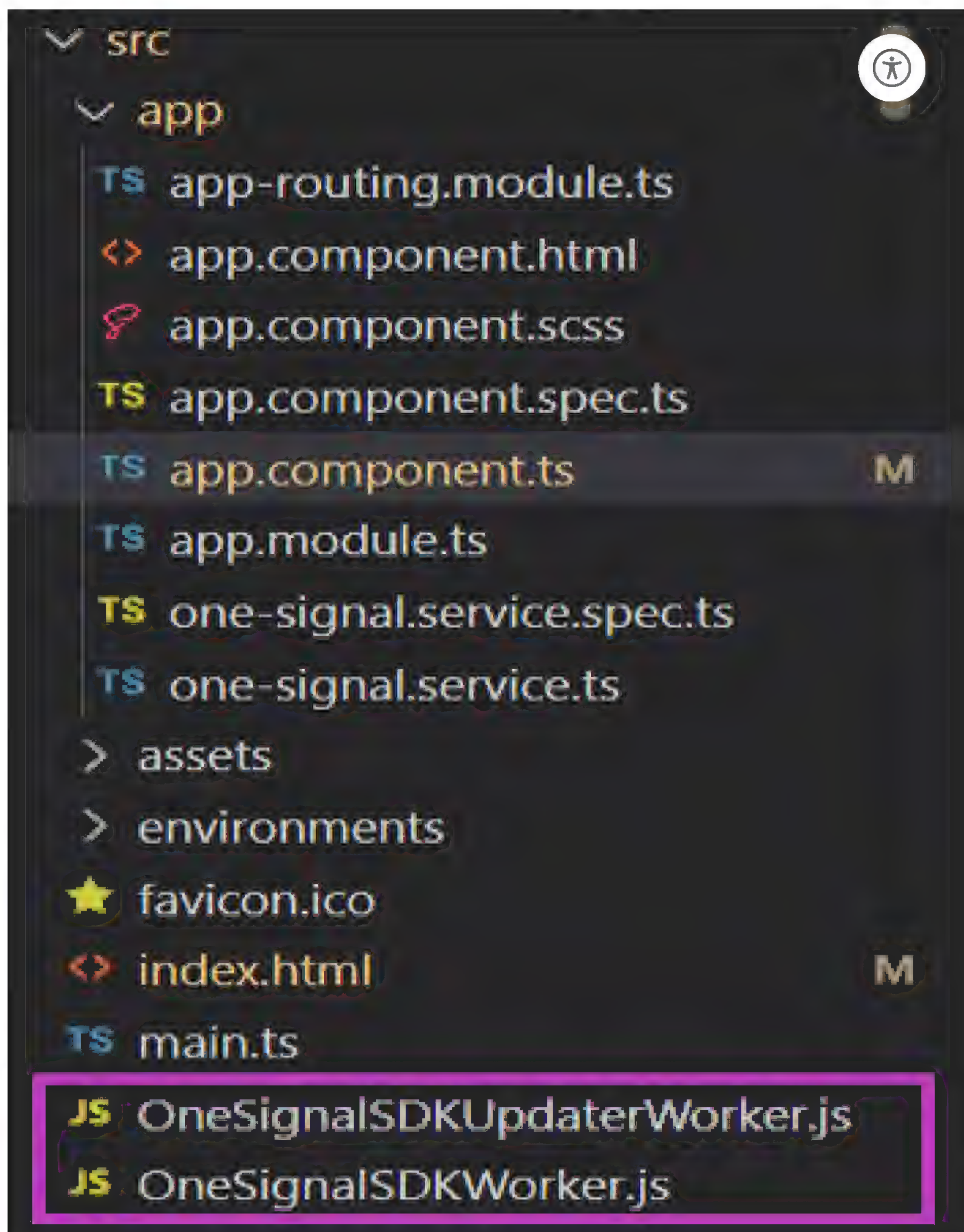


Figura2: Inserindo arquivos SDK no src

Depois de inserir os arquivos SDK em seu projeto Angular, você precisa tornar seu componente Angular ciente do pacote OneSignal NPM. Para fazer isso, navegue até o componente em que deseja usar

o pacote OneSignal NPM. Para este exemplo, estou usando o app.component.ts arquivo porque é o primeiro compone que meu aplicativo carregará.

Na parte superior do arquivo escolhido, importe o OneSignalService pacote OneSignal-ngx npm.

```
import { OneSignalService } from 'onesignal-ngx';
```

Figura3: Importando o OneSignalService

Agora, é hora de declarar o serviço OneSignal em seu construtor para que você possa usar o serviço para inicializar o OneSignal em nosso aplicativo. Cole o appldque você copiou anteriormente no espaço onde diz “YOUR-ONESIGNAL-APP-ID”.

O seu app.component.ts deve ficar assim:

```
@Component({
      selector: 'app-root',
      templateUrl: './app.component.html',
      styleUrls: ['./app.component.css']
})
export class AppComponent {
    title = 'angular-example-app';
    constructor(private oneSignal: OneSignalService) {
        this.oneSignal.init({
            appId: "YOUR-ONESIGNAL-APP-ID",
        });
    }
}
```

Figura 4: App.component.ts

Gerando Notificações

Permitir notificações por push da Web

Execute o aplicativo Angular e visite seu site. Você deve ver o seguinte prompt aparecer após o intervalo de atraso de tempo escolhido:

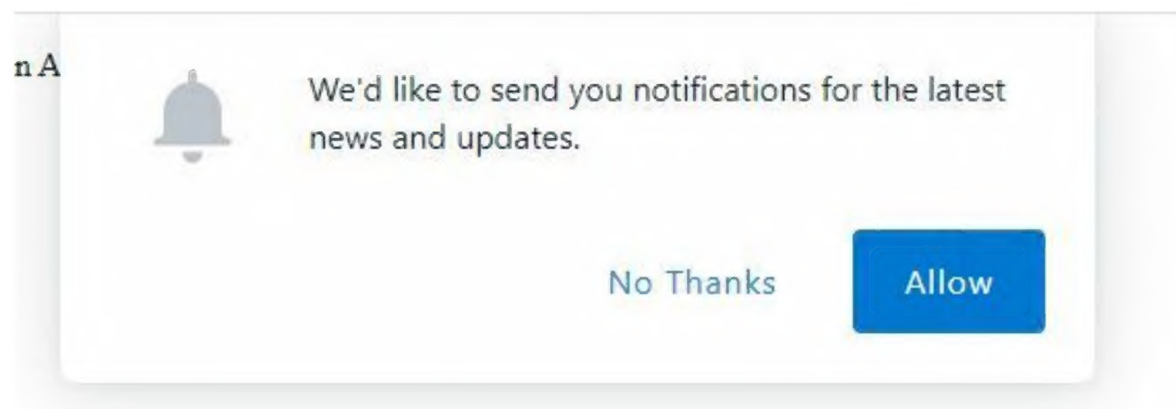


Figura 5: Prompt de retorno



Clique no botão azul Permitir para ativar as notificações push no seu navegador.

Enviar notificações por push da Web

É hora de enviar sua primeira notificação push da web! Para fazer isso, faça login na sua conta OneSignal e navegue até a guia Dashboard . Na página do painel, clique no botão azul que diz New Push .

Você será redirecionado para uma nova janela que permitirá personalizar sua notificação por push. Em Público , certifique-se de que Enviar para usuários inscritos esteja selecionado. Em seguida, crie sua mensagem adicionando o título da mensagem, o conteúdo e uma imagem. Como esta é a primeira notificação que seus assinantes receberão, você pode optar por criar uma mensagem simples de boas-vindas para confirmar que eles foram inscritos e reforçar o valor que as notificações fornecerão.

Na seção Cronograma de entrega , selecione Imediatamente e Enviar para todos ao mesmo tempo para enviar a todos os seus assinantes atuais do web push. Caso você tenha acabado de fazer a configuração da sua conta OneSignal, é muito provável que você seja o primeiro e único assinante. Se seu aplicativo ou site tiver tráfego intenso e outros usuários já optaram por receber notificações push, convém selecionar Enviar para segmentos específicos para testar sua mensagem em um público específico. Quando estiver pronto para enviar sua mensagem, clique no botão azul Revisar e Enviar na parte inferior da tela.

Um pequeno pop-up aparecerá para você revis sua mensagem. Quando estiver de acordo, irá clicar no botão azul de Enviar mensagens. Você deve receber uma notificação web push no seu dispositivo!

Atividade Extra

Para se aprofundar no assunto desta aula assista o vídeo:

Canal: Código Fonte TV

Título para buscar no youtube: **Firebase // Dicionário do Programador**

Referência Bibliográfica

MADUSANKA, Isuru. **Busy programmers guide to Firebase with Android**, 2013.

PAN, Daniel. 2016. **Firebase Tutorial.** October, 2016.

STONEHEM, Bill. **Google Android Firebase: Learning the Basics Paperback**, 2016.

Ir para exercício